REMARKS

In the Office Action mailed 05/22/2007, the Examiner has rejected Claims 1-51 under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The Examiner has specifically taken issue with the following language from Claims 1, 7, 18, 24, 35 and 41 as being indefinite: "more strongly." In the Amendment filed 08/22/2007, applicant respectfully asserted that such claim language is to be read according to the plain and ordinary meaning thereof, in view of dictionary definitions, etc. The Examiner, however, has argued that "it is uncertain what the association is stronger than." In response, applicant respectfully asserted that the association is stronger than it would be without the modification of the set of rules.

In the Office Action mailed 11/01/2007, has removed the rejection of Claims 1-51 under 35 U.S.C. 112, second paragraph, but has responded to applicant's above arguments. In particular, the Examiner has argued that applicant's above arguments are "not clear from the claim language," and that "it is not clear that the external program calls are more strongly associated with malicious computer program activity as compared to without the modifications." The Examiner has also argued that "[i]t could be more strongly associated with malicious computer program activity than the primary set of external program calls" such that "the scope of 'more strongly' cannot be ascertained."

Applicant respectfully disagrees. For example, with respect to the independent claims, applicant clearly claims "modifying said set of rules **such that** said at least one secondary set of one or more external program calls are more strongly associated with malicious computer program activity" (see this or similar, but not necessarily identical language in the independent claims-emphasis added), as claimed. Therefore, it is clear that applicant's claimed "said at least one secondary set of one or more external program calls are more strongly associated with malicious computer program activity" (emphasis added), as claimed, is definite.

The Examiner has rejected Claims 1, 2, 8-10, 13, 14, 17, 18, 19, 25-27, 30, 34, 35, 36, 42-44, 47, 48 and 51-53 under 35 U.S.C. 102(e) as being anticipated by van der Made (U.S. Patent No. 7,093,239). Applicant respectfully disagrees with such rejection, especially in view of the amendments made hereinabove to each of the independent claims. Specifically, applicant has amended the independent claims to at least substantially include the subject matter of former dependent Claims 2 et al. and 14 et al.

With respect to independent Claims 1, 18 and 35, the Examiner has relied on Col. 6, lines 12-24; and Col. 11, lines 46-60 from the van der Made reference to make a prior art showing of applicant's claimed "secondary set identifying code operable to identify, within said stream, at least one secondary set of one or more external program calls associated with said primary set of one or more external program calls" (see this or similar, but not necessarily identical language in the independent claims).

Applicant respectfully points out that the excerpts from the van der Made reference relied upon by the Examiner merely teach "extracting a behavior pattern and sequence from a modified, new, unknown or suspect program," and that "[t]he behavior pattern is preferably used to analyze the behavior of the unknown program to determine if the behavior of the unknown program is malicious" (Col. 6, lines 13-17 – emphasis added). The excerpts from van der Made also teach that the "ABM engine then analyzes the first executable program and finds that its behavior pattern is altered in a manner indicating that a virus is active" (Col. 11, lines 57-59 – emphasis added).

However, applicant respectfully asserts that only generally disclosing that "[t]he behavior pattern is preferably used to analyze the behavior of the unknown program," as in van der Made, does not specifically meet a "secondary set of identifying code operable to identify, within said stream, at least one secondary set of one or more external program calls associated with said primary set of one or more external program calls" (emphasis added), particularly where the "primary set of one or more external program calls match[es] one or more rules indicative of malicious computer program activity from among a set of rules" (emphasis added), in the context claimed by applicant.

Furthermore, applicant respectfully points out that detecting active viruses based on whether an executable program's behavior pattern is altered, as in van der Made, clearly fails to teach the use of a "secondary set of identifying code operable to identify, within said stream, at least one secondary set of one or more external program calls associated with said primary set of one or more external program calls" (emphasis added), where the "primary set of one or more external program calls match[es] one or more rules indicative of malicious computer program activity from among a set of rules" (emphasis added), in the context claimed by applicant. Simply nowhere in the excerpts relied on by the Examiner is there any teaching or suggestion of a "secondary set of one or more external program calls associated with said primary set of one or more external program calls," as claimed.

In the Office Action mailed 11/01/2007, the Examiner has argued that "Made discloses pattern identifying code that can identify program calls associated with malicious activity and are also associated with another set of program calls such as ones that are content destructive since these calls are calls that are made as a result of the first set of calls detected by patterns (6:43-63)."

Applicant respectfully disagrees and asserts that Col. 6, lines 43-63 in van der Made merely discloses that "the analysis procedure specifically targets infection methods such as, but not limited to, the insertion of code to other executables or documents, submitting code to other applications to be transmitted or stored, insertion of code into high memory blocks and the modification of memory control blocks," and that "the analysis method further look[s] for destructive content, such as, but not limited to, functions that overwrite disk areas or the BIOS ROM, or delete files or directories."

Clearly, the excerpts from van der Made merely teach targeting particular infection methods, and separately looking for destructive content, which does not even suggest "identifying code that can identify program calls associated with malicious activity and are also associated with another set of program calls such as ones that are

content destructive" (emphasis added), as the Examiner has noted. To this end, the excerpt from van der Made relied on by the Examiner simply does not teach a "secondary set of identifying code operable to identify, within said stream, at least one secondary set of one or more external program calls **associated with said primary set of one or more external program calls**" (emphasis added), where the "primary set of one or more external program calls match[es] one or more rules indicative of malicious computer program activity from among a set of rules" (emphasis added), in the context claimed by applicant.

Still with respect to independent Claims 1, 18 and 35, the Examiner has again relied on Col. 6, lines 12-24; and Col. 11, lines 46-60 from the van der Made reference to make a prior art showing of applicant's claimed "modifying code operable to modify said set of rules such that said at least one secondary set of one or more external program calls are more strongly associated with malicious computer program activity" (see this or similar, but not necessarily identical language in the independent claims).

Applicant respectfully points out that the excerpts from the van der Made reference relied upon by the Examiner merely teach "extracting a behavior pattern and sequence from a modified, new, unknown or suspect program," and that "[t]he behavior pattern is preferably used to analyze the behavior of the unknown program to determine if the behavior of the unknown program is malicious" (Col. 6, lines 13-17 – emphasis added). Such excerpts from van der Made also teach that the "ABM engine then analyzes the first executable program and finds that its behavior pattern is altered in a manner indicating that a virus is active" (Col. 11, lines 57-59 – emphasis added).

However, applicant respectfully asserts that analyzing "the behavior pattern of the unknown program," and detecting active viruses based on whether an executable program's behavior pattern is altered, as in van der Made, clearly fail to teach "modifying code operable to modify said set of rules such that said at least one secondary set of one or more external program calls are more strongly associated with malicious computer program activity," (emphasis added), as claimed by applicant, particularly where the

"rules [are] indicative of malicious computer program activity," in the context claimed. Simply nowhere in the excerpts from the van der Made reference relied on by the Examiner is there any teaching or suggestion to "modify said set of rules," as claimed by applicant.

In the Office Action mailed 11/01/2007, the Examiner has argued that "Made discloses modifying the behavior patterns as new malicious behavior is detected and as more malicious behavior is detected it associated the patterns and the calls that fall within the pattern more closely with the malicious activity (6:25-43)."

Applicant respectfully disagrees and asserts that Col. 6, lines 25-43 in van der Made simply teaches that "a virtual machine is used to generate a behavior pattern and a sequence," and that "[t]he generated behavior pattern does not change significantly between version updates, but does change dramatically when a virus infects a program." However, simply disclosing that a behavior pattern changes when a virus infects a program, as in van der Made, does not even suggest that "as more malicious behavior is detected it associated the patterns and the calls that fall within the pattern more closely with the malicious activity" (emphasis added), as the Examiner has noted. Furthermore, a behavior pattern that changes when a virus infects a program, as in van der Made, does not teach "modifying code operable to modify said set of rules such that said at least one secondary set of one or more external program calls are more strongly associated with malicious computer program activity," (emphasis added), as claimed by applicant, particularly where the "rules [are] indicative of malicious computer program activity," in the context claimed.

The Examiner is reminded that a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described in a single prior art reference. *Verdegaal Bros. v. Union Oil Co. Of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). Moreover, the identical invention must be shown in as complete detail as contained in the claim. *Richardson v. Suzuki Motor*

*Co.*868 F.2d 1226, 1236, 9USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim.

This criterion has simply not been met by the above reference excerpt(s), as noted above. Nevertheless, despite such paramount deficiencies and in the spirit of expediting the prosecution of the present application, applicant has amended each of the independent claims to further distinguish applicant's claim language from the above reference by incorporating the subject matter of former dependent Claims 2 et al. and 14 et al.

With respect to the subject matter of former Claims 2 et al. (now at least substantially incorporated into each of the independent claims), the Examiner has relied on Col. 6, lines 12-24 (excerpted below) from the van der Made reference to make a prior art showing of applicant's claimed technique "wherein one of said at least one secondary set of one or more external program calls precedes said primary set of one or more external program calls within said stream of external program calls" (see this or similar, but not necessarily identical language in the independent claims).

> "Preferred implementations of the analytical behavior method (ABM) proceed by extracting a behavior pattern and sequence from a modified, new, unknown or suspect program. The behavior pattern is preferably used to analyze the behavior of the unknown program to determine if the behavior of the unknown program is malicious. Identification of malicious behavior in this manner allows identification of virus carrying files prior to infection of the host computer system. The behavior pattern can also be stored in a database and the virtual machine can subsequently analyze the behavior of the program following modification to determine if its functionality has been modified in a suspect (malicious) manner. This provides post-infection analysis." (Col. 6, lines 12-24 - emphasis added)

Applicant respectfully points out that the excerpt from the van der Made reference relied upon by the Examiner merely teaches "extracting a behavior pattern and sequence from a modified, new, unknown or suspect program," and that "[t]he behavior pattern is preferably used to analyze the behavior of the unknown program to determine if the behavior of the unknown program is malicious" (Col. 6, lines 13-17 – emphasis added).

However, applicant respectfully asserts that only generally disclosing that "[t]he behavior pattern is preferably used to analyze the behavior of the unknown program," as in van der Made, fails to specifically disclose a technique "wherein one of said at least one secondary set of one or more external program calls **precedes** said primary set of one or more external program calls within said stream of external program calls" (emphasis added), as claimed by applicant.

In the Office Action mailed 11/01/2007, the Examiner has failed to respond to applicant's arguments with respect to applicant's claimed technique "wherein one of said at least one secondary set of one or more external program calls **precedes** said primary set of one or more external program calls within said stream of external program calls" (emphasis added), as claimed by applicant. Thus, a notice of allowance or specific prior art showing of each of the foregoing claim elements, in combination with the remaining claimed features, is respectfully requested.

In addition, with respect to the subject matter of former Claims 14 et al. (now at least substantially incorporated into each of the independent claims), the Examiner has relied on Col. 11, lines 46-59 (excerpted below) from the van der Made reference to make a prior art showing of applicant's claimed technique "wherein said set of rules is modified to include a new rule corresponding to said secondary set of one or more external program calls, said new rule thereafter being used in addition to other rules within said set of rules" (see this or similar, but not necessarily identical language in the independent claims).

> "Post-infection detection
>
> Post-infection detection takes place in cases when initial infection is missed by pre-infection detection. A virus could be missed by pre-infection detection when it does not perform any viral function on first execution and does not modify interrupt vectors that point to an infection routine. This is the case with so-called slow infectors and similarly behaving malignant code. In post-infection detection the virus is caught the moment it attempts to infect the first executable on the PC. The file hook mechanism detects this attempted change to an executable (including documents). The ABM engine then analyzes the first executable program and finds that its behavior pattern is altered

in a manner indicating that a virus is active." (Col. 11, lines 46-59 - emphasis added)

Applicant respectfully points out that the excerpt from the van der Made reference relied upon by the Examiner merely teaches detecting a virus in an executable program if the program's "behavior pattern is altered in a manner indicating that a virus is active" (Col. 11, lines 58-59 – emphasis added).

However, applicant respectfully asserts that detecting an active virus in a program because the program's behavior pattern is altered, as in van der Made, clearly does not teach that a "set of rules is modified to include a new rule corresponding to said secondary set of one or more external program calls," especially where "said new rule thereafter [is] used in addition to other rules within said set of rules" (emphasis added), as claimed by applicant.

In the Office Action mailed 11/01/2007, the Examiner has failed to respond to applicant's arguments with respect to applicant's claimed technique where a "set of rules is modified to include a new rule corresponding to said secondary set of one or more external program calls," especially where "said new rule thereafter [is] used in addition to other rules within said set of rules" (emphasis added), as claimed by applicant. Thus, a notice of allowance or specific prior art showing of each of the foregoing claim elements, in combination with the remaining claimed features, is respectfully requested.

Again, the foregoing anticipation criterion has simply not been met by the above reference excerpt(s), as noted above. Thus, a notice of allowance or specific prior art showing of each of the foregoing claim elements, in combination with the remaining claimed features, is respectfully requested.

Applicant further notes that the prior art is also deficient with respect to the dependent claims. For example, with respect to Claims 17 et al., the Examiner has relied on Col. 12, lines 26-41 (excerpted below) from the van der Made reference to make a prior art showing of applicant's claimed technique "wherein said set of rules is subject to

a validity check after modification to determine if said set of rules is more effectively detecting malicious computer program activity."

> "In tests of a prototype implementation ABM system, the combination of pre-infection (96%) and post-infection detection (4%) resulted in 100% detection of all known viral techniques, using a combination of new, modified and well-known viruses. Other methods detected only 100% of known viruses and scored as low as 0% for the detection of new, modified and unknown viruses. No exact figure can be quoted for tests involving signature scanner based products. The results for such products are a direct representation of the mix of known, modified and new, unknown viruses; e.g. if 30% of the virus test set is new, modified or unknown then the final score reflected close to 30% missed viruses. No such relationship exists for the implementations of preferred aspects of the present system, where the detection efficiency does not appreciably vary for alterations of the presented virus mix." (Col. 12, lines 26-41 - emphasis added)

Applicant respectfully points out that the excerpt from the van der Made reference relied upon by the Examiner merely discloses "tests of a prototype implementation ABM system, the combination of pre-infection (96%) and post-infection detection (4%) resulted in 100% detection of all known viral techniques, using a combination of new, modified and well-known viruses" (Col. 12, lines 26-41 – emphasis added).

However, applicant respectfully asserts that "tests of a prototype implementation ABM system," as in van der Made, clearly do not teach that a "set of rules is subject to a validity check after modification to determine if said set of rules is more effectively detecting malicious computer activity" (emphasis added), as claimed by applicant. Simply nowhere in the excerpt from the van der Made reference relied on by the Examiner is there any teaching or suggestion of a "validity check after modification [of said set of rules]," as claimed by applicant.

In the Office Action mailed 11/01/2007, the Examiner has argued that "Made discloses a test on a prototype that analyzed the validity of the rules and Made discloses that validity is checked when patterns are detected in order to ensure no false alarms (10:52-11:7)."

"The resulting behavior pattern is: 24AA952F2A244905

The behavior pattern contains flags that indicate that the user has not had the opportunity to interact with this process through user input (the userInput flag is not set). The sequencer contains the order in which the bits were set, identifying the infection sequence shown above. Therefore this observed behavior is most likely viral.

Many viruses are encrypted, polymorphic or use `tricks` to avoid detection by signature scanners. Wherever such `tricks` are used, the behavior pattern points more obviously towards a virus since such tricks are not normally used in normal applications. In any case, preferred implementations of the present invention require that an infection procedure be present to trigger a virus warning to avoid false positive warnings. Encrypted viruses are no problem, because the execution of the code within the virtual machine, which generates the behavior pattern, effectively decrypts any encrypted or polymorphic virus, as it would in a physical PC environment. Because all parts of the virtual computer are virtualized in preferred embodiments, and at no time is the virtualized program allowed to interact with the physical computer, there is no chance that viral code could escape from the virtual machine and infect the physical computer." (Col. 10, line 52 - Col. 11, line 7 - emphasis added).

Applicant respectfully disagrees and asserts that the excerpt from the van der Made reference relied upon by the Examiner merely teaches that "[t]he behavior pattern contains flags that indicate that the user has not had the opportunity to interact with this process through user input" and that "preferred implementations of the present invention require that an infection procedure be present to trigger a virus warning to avoid false positive warnings" (emphasis added).

However, teaching that the behavior pattern contains flags indicating that the user has not had the opportunity to interact with this process, in addition to teaching that an infection procedure is required to be present to trigger a virus warning, as in van der Made, simply fails to suggest that a "set of rules is subject to a validity check after modification to determine if said set of rules is more effectively detecting malicious computer activity" (emphasis added), as claimed by applicant. Clearly, a flag in the behavior pattern indicating that the user has not interacted with the process, as in van der

Made, simply fails to even suggest that a "set of rules is subject to a validity check after modification" (emphasis added), as claimed by applicant.

Further, with respect to Claim 52, the Examiner has relied on Col. 10, line 18-Col. 11, line 23; and Col. 12, lines 26-41 from the van der Made reference to make a prior art showing of applicant's claimed "applying high level rules to the modified set of rules, and promoting said modified set of rules from a temporary set to a permanent set based on the application of the high level rules to the modified set of rules."

Applicant respectfully asserts that the excerpts from the van der Made reference relied upon by the Examiner merely teach that "[t]he sequencer contains the order in which the bits were set, identifying the infection sequence shown above" (Col. 10, lines 55-57). Further, the excerpts teach that "[t]he change detection module compares existing files at 6 levels to determine if the file was analyzed previously" (Col. 11, lines 8-9 – emphasis added). Additionally, the excerpts teach that "[i]n tests of a prototype implementation ABM system, the combination of pre-infection (96%) and post-infection detection (4%) resulted in 100% detection of all known viral techniques, using a combination of new, modified and well-known viruses" (Col. 12, lines 26-30 – emphasis added).

However, identifying the infection sequence, comparing files to determine if the file was previously analyzed, and teaching that the combination of pre-infection and post-infection detection resulted in 100% detection of all known viral techniques, as in van der Made, simply fails to suggest "applying high level rules to the modified set of rules, and promoting said modified set of rules from a temporary set to a permanent set based on the application of the high level rules to the modified set of rules" (emphasis added), as claimed by applicant. Clearly, pre-infection and post-infection detection of viral techniques, in addition to identifying an infection sequence, and determining if a file was previously analyzed, as in van der Made, simply fails to even suggest "**promoting** said **modified** set of rules from a **temporary set** to a **permanent set** based on the application

of the high level rules to the modified set of rules" (emphasis added), as claimed by applicant.

In addition, with respect to Claim 53, the Examiner has relied on Col. 10, line 18-Col. 11, line 23; and Col. 12, lines 26-41 from the van der Made reference to make a prior art showing of applicant's claimed "determining whether said modified set of rules decrease malicious network traffic, and promoting said modified set of rules from a temporary set to a permanent set if it is determined that said modified set of rules decrease said malicious network traffic."

Applicant respectfully asserts that the excerpts from the van der Made reference relied upon by the Examiner merely teach that "[t]he sequencer contains the order in which the bits were set, identifying the infection sequence shown above" (Col. 10, lines 55-57). Further, the excerpts teach that "[t]he change detection module compares existing files at 6 levels to determine if the file was analyzed previously" (Col. 11, lines 8-9 – emphasis added). Additionally, the excerpts teach that "[i]n tests of a prototype implementation ABM system, the combination of pre-infection (96%) and post-infection detection (4%) resulted in 100% detection of all known viral techniques, using a combination of new, modified and well-known viruses" (Col. 12, lines 26-30 – emphasis added).

However, identifying the infection sequence, comparing files to determine if the file was previously analyzed, and teaching that the combination of pre-infection and post-infection detection resulted in 100% detection of all known viral techniques, as in van der Made, simply fails to suggest "malicious network traffic," much less "determining whether said modified set of rules decrease malicious network traffic, and promoting said modified set of rules from a temporary set to a permanent set if it is determined that said modified set of rules decrease said malicious network traffic" (emphasis added), as claimed by applicant. Clearly, pre-infection and post-infection detection of viral techniques, in addition to identifying an infection sequence, and determining if a file was previously analyzed, as in van der Made, simply fails to even suggest "**promoting said**

modified set of rules from a temporary set to a permanent set if it is determined that said modified set of rules **decrease said malicious network traffic**" (emphasis added), as claimed by applicant.

Again, since the above anticipation criterion has simply not been met by the above reference excerpt(s), as noted above, a notice of allowance or specific prior art showing of each of the foregoing claim elements, in combination with the remaining claimed features, is respectfully requested.

Still yet, applicant brings to the Examiner's attention the subject matter of new Claim 54 below, which is added for full consideration:

> "further comprising promoting code operable to determine whether said modified set of rules slows malware propagation, and to promote said modified set of rules from a temporary set to a permanent set if it is determined that said modified set of rules slows said malware propagation" (see Claim 54).

Again, a notice of allowance or a proper prior art showing of all of applicant's claim limitations, in combination with the remaining claim elements, is respectfully requested.

Thus, all of the independent claims are deemed allowable. Moreover, the remaining dependent claims are further deemed allowable, in view of their dependence on such independent claims.

In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at (408) 505-5100. The

Commissioner is authorized to charge any additional fees or credit any overpayment to Deposit Account No. 50-1351 (Order No. NAI1P489).

Respectfully submitted,
Zilka-Kotab, PC

/KEVINZILKA/

Kevin J. Zilka
Registration No. 41,429

P.O. Box 721120
San Jose, CA 95172-1120
408-505-5100